Password Security & Authentication Analysis

1] How Passwords Are Stored

- In websites, when a user creates a password, it is passed through a hashing algorithm which converts it into a fixed-length string called a hash.
- During login, the entered password is hashed again and compared with the stored hash. If both hashes match, access is granted.
- Since hashing is a one-way process, the original password cannot be retrieved from the hash, which improves security even if the database is compromised.
- Forgot password does not recover the old password, it verifies the user using a temporary token.
- This token is time-limited and can be used only once. When the user opens the reset link, the token is verified and the user is allowed to create a new password.

2] Hashing v/s Encryption

Hashing	**Encryption**
One-way	Two-way
Cannot be reversed	Can be decrypted
No key used	Uses secret key
Used for passwords	Used for data
Output always same	Output varies with key

Passwords are hashed instead of encrypted because hashing does not require storing any secret key, which prevents password recovery even if the database is compromised.

3] Hash Types & Strength Analysis

MD5 (Very Weak)

Characteristics:

- 128-bit hash
- Very fast
- No salt
- Easily crackable

Problems:

- Vulnerable to dictionary attacks
- Vulnerable to rainbow tables
- Collisions exist

MD5 is broken and insecure.

SHA-1 (Weak)

Characteristics:

- 160-bit hash
- Slightly stronger than MD5
- Still very fast
- Collision attacks possible

Security Status:

- Officially deprecated
- No longer recommended

SHA-1 is also considered insecure.

SHA-256 (Moderate)

Characteristics:

- Part of SHA-2 family
- 256-bit hash
- Strong mathematically
- Still fast

Problems:

- Fast speed helps attackers
- No built-in salt

Used in:

- file integrity checks
- blockchain
- digital signatures

Better than MD5/SHA-1 but not ideal alone for passwords.

bcrypt (Strong & Recommended)

Characteristics:

- Adaptive hashing algorithm
- Built-in salt
- Slow by design
- Configurable cost factor

Advantages:

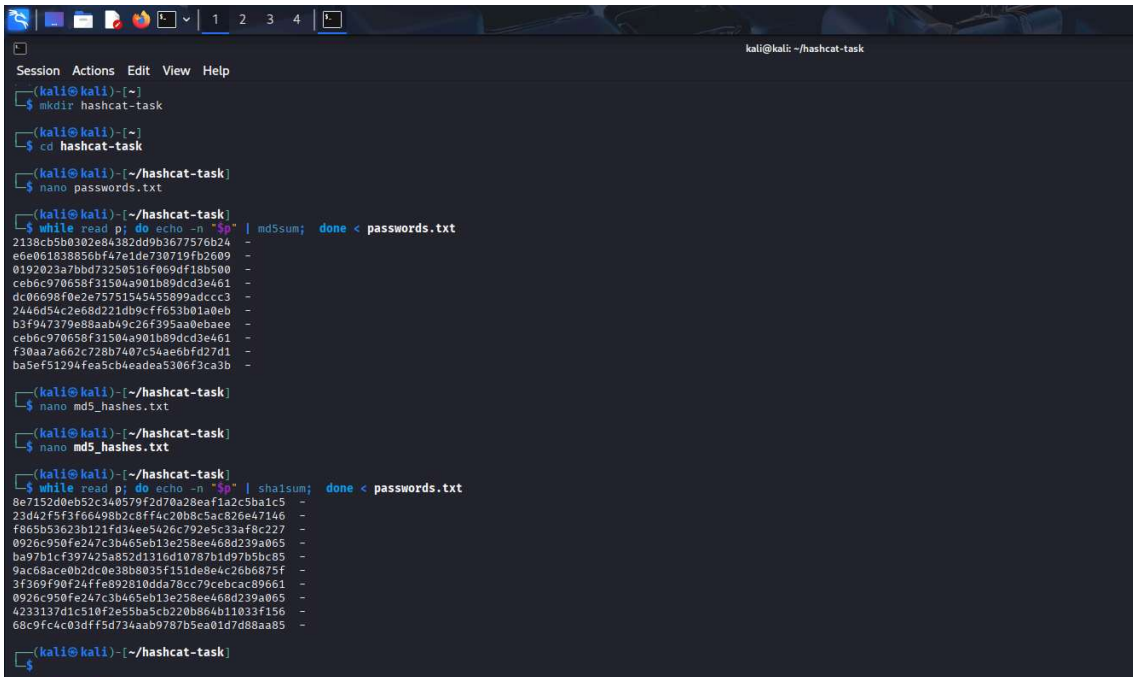
- Resistant to brute-force attacks
- Resistant to GPU cracking
- Automatically adds salt

Best choice for passwords.

Salt = random value added to password before hashing.
password123 + random_salt → hash

4] Password Hash Generation & Cracking (Using Hashcat)

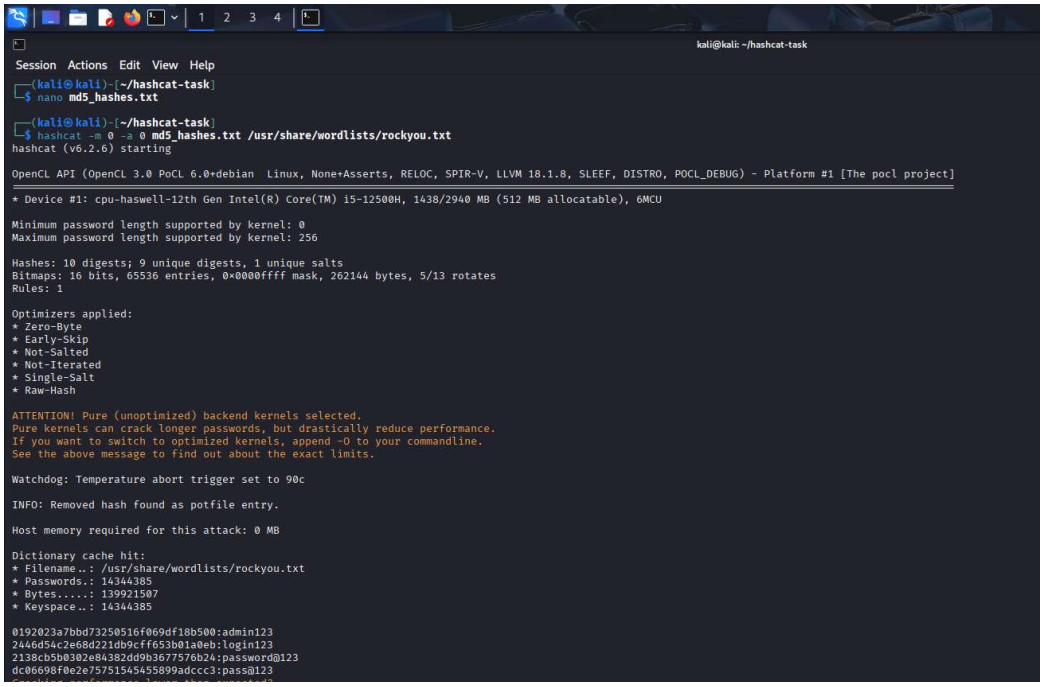
Generation of MD5 and SHA-1 Hashes



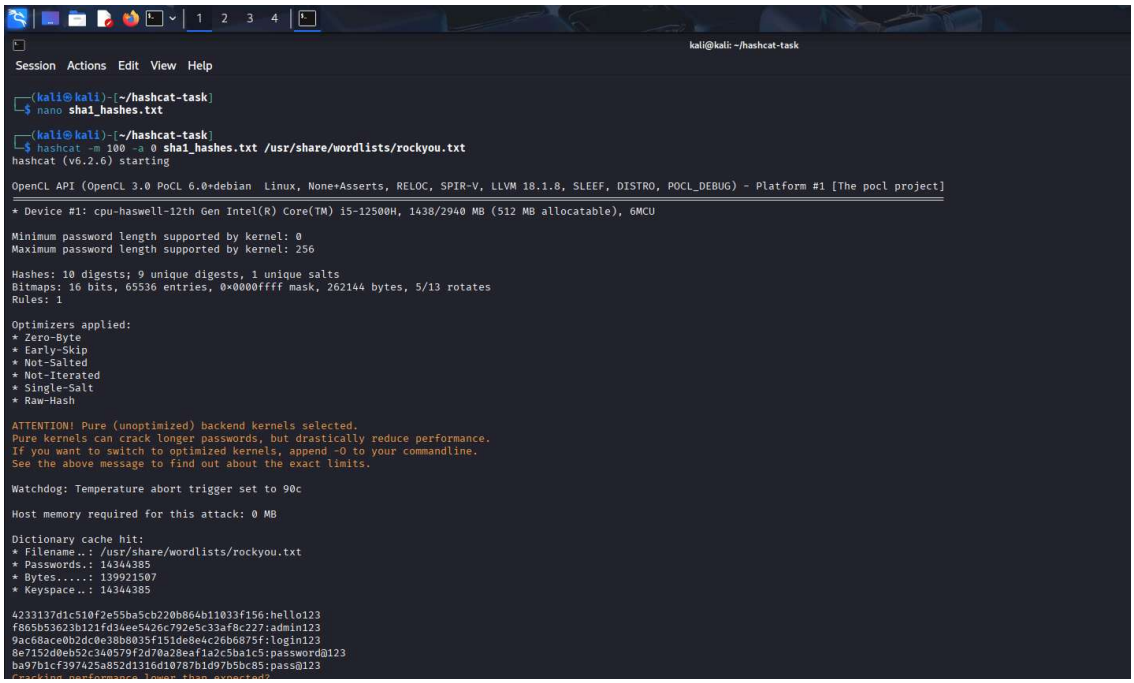
Hash Table:

No.	Password	MD5 Hash	SHA-1 Hash
1	password123	2138cb5b0302e84382dd9b3677576b24	8e7152d0eb52c340579f2d70a28eaf1a2c5ba1c5
2	admin@123	e6061838856bf47e1de730719fb2609	23d42f5f3f66498b2c8ff4c20b8c5ac826e47146
3	admin123	0192023a7bbd73250516f069df18b500	f865b53623b121fd34ee5426c792e5c33af8c227
4	test@123	ceb6c970658f31504a901b89dcd3e461	0926c950fe247c3b465eb13e258e468d239a065
5	pass@123	dc06698f0e2e75751545455899adccc3	ba97b1cf397425a852d1316d10787b1d97b5bc85
6	login123	2446d54c2e68d221db9cff653b01a0eb	9ac68ace0b2dc0e38b8035f151de8e4c26b6875f
7	abcd@123	b3f947379e88aab49c26f395aa0ebaee	3f369f90f24ffe892810dda78cc79cebcac89661
8	test123	ceb6c970658f31504a901b89dcd3e461	0926c950fe247c3b465eb13e258e468d239a065
9	hello123	f30aa7a662c728b7407c54ae6bfd27d1	4233137d1c510f2e55ba5cb220b864b11033f156
10	user@123	ba5ef51294fea5cb4eadea5306f3ca3b	68c9fc4c03dff5d734aab9787b5ea01d7d88aa85

MD5 Hash cracking



SHA-1 Hash Cracking



Observations:

- Multiple MD5 and SHA-1 hashes were successfully cracked using Hashcat through dictionary attacks.
- All weak passwords were recovered quickly because they were present in commonly used wordlists.
- This demonstrates that fast hashing algorithms such as MD5 and SHA-1 provide poor password protection and are vulnerable to offline cracking attacks.

John The Ripper:

- Free password cracking software tool.
- Originally developed for the Unix operating system; can run on fifteen different platforms (eleven of which are architecture-specific versions of Unix, DOS, Win32, BeOS, and OpenVMS).

5] Understand brute force vs dictionary attacks.

Dictionary Attack:

- A dictionary attack is a password-cracking method that uses a predefined list of commonly used passwords.
- The attacker uses a wordlist (example: rockyou.txt)
- Each word from the list is tried as a password
- The hash of each word is compared with the target hash
- If a match is found, the password is cracked
- Uses real-world leaked passwords

Brute Force Attack:

- A brute force attack tries every possible combination of characters until the correct password is found.
- Each combination is hashed and compared
- Does not rely on wordlists

Password length	Time (approx)
4 characters	seconds
6 characters	minutes
8 characters	days
10+ characters	years

Why Weak Passwords Fail Easily

- They lack randomness
- They follow human behavior patterns
- They appear in leaked datasets
- They are short and simple
- They do not resist automated attack tools

Multifactor Authentication (MFA):

Multi-Factor Authentication (MFA) is a security mechanism that requires users to verify their identity using two or more authentication factors instead of just a password.
It usually includes:
Something you know → Password
Something you have → OTP / Authenticator app
Something you are → Fingerprint / Face ID

Why passwords are not enough:

- Passwords can be guessed or cracked using brute-force and dictionary attacks
- Many users reuse the same password across multiple websites
- Passwords can be leaked through data breaches and phishing attacks
- Once compromised, attackers can fully access the account

Importance of MFA:

- Adds an extra security layer beyond passwords
- Blocks attackers even if the password is cracked
- Prevents unauthorized logins from unknown devices
- Protects against phishing and credential-stuffing attacks
- Requires physical access (OTP/biometric) to complete login
- Significantly reduces account takeover risks

Interview Questions:
What is hashing?
Difference between hashing and encryption?
What is brute force attack?
Why is MFA important?
What makes a strong password?